External Memory Algorithms for Reducing Data Movement

Ed D'Azevedo (dazevedoef@ornl.gov) and Judith Hill
Oak Ridge National Laboratory

Background

Several reports [2, 6, 14] on the technology path towards exascale computing have presented two possible technology paths or "swim-lanes" for an exascale machine: (i) about one million nodes, each node consists of many-core processors with performance at 1 TFlops/s; (ii) about 10,000 nodes, each node has high performance accelerators (perhaps based on graphics processing units (GPU)) with performance at about 100 TFlops. The technology road map [14, Chapter 6] also compares different memory technologies and future nodes will likely have user managed caches (such as shared memory on GPU) and non-volatile (NV) memory based on phase-change or flash technology. NV memory may be used to augment main memory or act as fast disk cache for storing check-point data. All these reports have emphasized that energy usage is the critical challenge and *the biggest energy cost is in data movement.* Dr William Harrod, the Director of Research in the Office of Advanced Scientific Computing Research (ASCR), summarizes the issue as:[1]

> ***It is not about the FLOPS. It is about data movement.***

This document attempts to address a gap in current thinking about applied mathematics for the exascale by proposing development of external memory (or out-of-core) algorithms for developing efficient solvers on accelerators and address the issue of reducing data movement for computing at the exascale.

External Memory Algorithms for Reducing Data Movement

The world's fastest Titan supercomputer at the National Center for Computational Sciences (top rank in TOP500) uses NVIDIA Kepler GPU to achieve a peak performance over 20 PetaFlops yet is also ranked third most energy efficient on the Green500 list. Each compute node on Titan has a 16-core AMD Interlagos processor with 32 GBytes of memory and an NVIDIA Kepler GPU with only 6 GBytes of device memory. Out-of-core algorithms are useful for solving large problems limited by only the total available memory and not by the smaller amount of GPU device memory. Similar to the major challenge in exascale computing, the hurdle to achieving high performance on a hybrid GPU and CPU configuration is the high cost of data movement between the CPU and GPU. The NVIDIA Titan GPU has 288 GBytes/sec of memory bandwidth on device memory but data transfer between the GPU device and CPU host is only about 6 GBytes/sec.

The analysis and implementation of out-of-core algorithms or external memory algorithms were developed in the early days of computing to solve large problems (stored on disk) on computers that had relatively small main memories. Hong and Kung [11] used a simple red/blue pebble game to analyse out-of-core algorithms to derive lower bounds for $n \times n$ matrix multiply and n-point Fast Fourier Transform (FFT) in S amount of memory. This simple model was later extended to the analysis of multi-level memory hierarchies by Savage [15]. Vitter has written surveys [1, 21] of external memory algorithms including sorting, isosurface extraction, and external memory graph algorithms. In particular, Toledo [17] has a chapter on out-of-core algorithms in numerical linear algebra that includes dense matrix multiplication, sparse Cholesky factorization, FFT, n-body computation, and dense eigensolver. Toledo has designed a recursive schedule for out-of-core dense LU factorization with partial pivoting that may be more efficient than left-looking block algorithm [16]. Similar ideas of recursive algorithms and block tiling have been shown to

[1] Presentation at the 2011 DOE Applied Mathematics Program Meeting is available at `http://www.csm.ornl.gov/workshops/applmath11/documents/talks/Harrod_Extreme_Scale.pdf`

be effective on other dense matrix algorithms [10]. There are several parallel out-of-core solvers for large dense matrix calculations. Toledo and Gustavson [18] have developed a solver for large scale QR and Singular Value Decomposition (SVD) for long slender matrices. POOCLAPACK [9] is an out-of-core extension of the PLAPACK [19] library for performing Cholesky and QR factorization that is based on a block partitioned algorithm. D'Azevedo and Dongarra [4] have developed an out-of-core extension to ScaLAPACK for LU, Cholesky, and QR factorization. Since partial pivoting is required in LU factorization, the ScaLAPACK algorithms schedule computations in block column panels. In all cases, parallel out-of-core algorithms for dense matrix computations have been shown to achieve high performance.

Relying on automatic data transfer by paging in the virtual memory system may not lead to high performance. The very long latency of transferring data between disk and CPU core memory made this research necessary to minimize the amount of data transfer. Most of the analysis assumed just two levels of memory, which is a small fast in-core memory and much slower disks with larger capacity. The long computation time in solving large problems also motivated the development of techniques for resilience in restarting time consuming calculations. As the size of computer memories increased and large problems are ported to run on distributed memory machines, the interest in out-of-core algorithms has waned.

Due to the very large problem sizes, some out-of-core calculations may take days to complete since solving a problem that is K times the size of available main memory would increase runtime by a factor of $K^{3/2}$. The check-point and restart capability is simplified by having the data on the disk. The sequence of operations or subroutine calls can be generated and written out [5]. This list of operations allows the driver to easily determine what needs to be restored or partial results that needs to be recomputed at restart. Out-of-core techniques for dense matrix factorization can be adapted for other memory hierarchies such as between CPU and NVRAM, or between CPU and accelerators. Thus out-of-core ideas are well matched to reducing costly data movement and in exploiting NVRAM technology for exascale computing.

A parallel external memory solver for dense complex matrices that uses two-dimensional block cyclic data distribution to be compatible with ScaLAPACK `PxGETRF` has been developed to take advantage of GPU acceleration and can solve problems larger than available device memory. The solver is used in the AORSA (All Orders Spectral Algorithm) fusion application [12, 13]. A mixed precision version of the solver with iterative refinement achieves about four times speedup over using only the 16 multi-core CPUs on each node. Table 1 shows the performance of AORSA on the Titandev development system using Fermi GPU processors. Out-of-core techniques for parallel Cholesky factorization with GPU acceleration that is compatible with ScaLAPACK `PxPOTRF` has also been developed for computing the view factor matrix in modeling radiation heat transfer [22] on the Keeneland GPU cluster [20]. External memory algorithms can also be applied to solving eigenvalue problems by first using block orthogonal similarity transformations to reduce an out-of-core symmetric matrix to narrow block tridiagonal or banded form that can fit entirely in fast memory [8]. A divide-and-conqueror algorithm [7, 3] can then be used.

matrix size	titandev nodes	Gflops/node	Total Tflops
395,523	153	376	57.6
395,523	253	326	82.4
789,507	561	352	197.7
789,507	903	285	257.5

Table 1: Performance of parallel mixed precision complex GPU solver on Titandev (Fermi GPU), 8 MPI tasks per node.

References

[1] James M Abello and Jeffrey Scott Vitter, eds., *External memory algorithms: DIMACS Workshop External Memory and Visualization, May 20-22, 1998*, vol. 50 of Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1999.

[2] Steve Ashby, Pete Beckman, Jackie Chen, Phil Colella, Bill Collins, Dona Crawford, Jack Dongarra, Doug Kothe, Rusty Lusk, Paul Messina, Tony Mezzacappa, Parviz Moin, Mike Norman, Robert Rosner, Vivek Sarkar, Andrew Siegel, Fred Streitz, Andy White, and Margaret Wright, *The opportunities and challenges of exascale computing, summary report of the Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee*, tech. report, Office of Science, Department of Energy, Fall 2010. Report available at `http://science.energy.gov/~/media/ascr/ascac/pdf/reports/Exascale_subcommittee_report.pdf`.

[3] Yihua Bai and Robert C. Ward, *A parallel symmetric block-tridiagonal divide-and-conquer algorithm*, Tech. Report 33, University of Tennessee, 2007.

[4] E. F. D'Azevedo and J. J. Dongarra, *The design and implementation of the parallel out-of-core ScaLAPACK LU, QR and Cholesky factorization routines*, Concurrency: Practice and Experience, 12 (2000), pp. 1481–1493.

[5] E. F. D'Azevedo and Piotr Luszczek, *A framework for check-pointed fault-tolerant out-of-core linear algebra*. Presented at the SIAM Conference on Computational Science and Engineering, Feb 2003, San Diego, CA.

[6] J. Dongarra, P. Beckman, Terry Moore, Patrick Aerts, Giovanni Aloisio, Jean:Claude Andre, David Barkai, Jean:Yves Berthou, Taisuke Boku, Bertrand Braunschweig, Franck Cappello, Barbara Chapman, Xuebin Chi, Alok Choudhary, Sudip Dosanjh, Thom Dunning, Sandro Fiore, Al Geist, Bill Gropp, Robert Harrison, Mark Hereld, Michael Heroux, Adolfy Hoisie, Koh Hotta, Yutaka Ishikawa, Zhong Jin, Fred Johnson, Sanjay Kale, Richard Kenway, David Keyes, Bill Kramer, Jesus Labarta, Alain Lichnewsky, Thomas Lippert, Bob Lucas, Barney Maccabe, Satoshi Matsuoka, Paul Messina, Peter Michielse, Bernd Mohr, Matthias Mueller, Wolfgang Nagel, Hiroshi Nakashima, Michael E. Papka, Dan Reed, Mitsuhisa Sato, Ed Seidel, John Shalf, David Skinner, Marc Snir, Thomas Sterling, Rick Stevens, Fred Streitz, Bob Sugar, Shinji Sumimoto, William Tang, John Taylor, Rajeev Thakur, Anne Trefethen, Mateo Valero, Aad van der Steen, Jeffrey Vetter, Peg Williams, Robert Wisniewski, and Kathy Yelick, *The international exascale software roadmap*, International Journal of High Performance Computer Applications, 25 (2011). The report is available at `http://www.exascale.org/mediawiki/images/2/20/IESP-roadmap.pdf`.

[7] Wilfried N. Gansterer, Robert C. Ward, Richard P. Muller, and William A. Goddard III, *Computing approximate eigenpairs of symmetric block tridiagonal matrices*, 2003.

[8] R. G. Grimes and H. D. Simon, *Solution of large, dense symmetric generalized eigenvalue problems using secondary storage*, ACM Transactions on Mathematical Software, 14 (1988), pp. 241–256.

[9] Brian C. Gunter, Wesley C. Reiley, and Robert A. van de Geijn, *Parallel out-of-core Cholesky and QR factorizations with POOCLAPACK*, in Proceedings of the 15th International Parallel and Distributed Processing Symposium, San Francisco, CA, April 23-27, 2001.

[10] F. Gustavson, *Recursion leads to automatic variable blocking for dense linear-algebra algorithms*, IBM Journal of Research and Development, 41 (1997), pp. 737–755.

[11] Jia-Wei Hong and H. T. Kung, *I/O complexity: The red-blue pebble game*, in Proceedings of the thirteenth annual ACM symposium on Theory of computing, STOC '81, New York, NY, USA, 1981, ACM, pp. 326–333.

[12] E. F. Jaeger, L. A. Berry, E. F. D'Azevedo, D. B. Batchelor, M. D. Carter, and H. Weitzner, *Advances in full-wave modeling of radio frequency heated, multidimensional plasmas*, Physics of Plasmas, 9 (2002), pp. 1873–1881.

[13] E. F. Jaeger, L. A. Berry, J. R. Myra, D. B. Batchelor, E. F. D'Azevedo, P. T. Bonoli, C. K. Philips, D. N. Smithe, D. A. D'Ippolito, M. D. Carter, R. J. Dumont, J. C. Wright, and R. W. Harvey, *Sheared poloidal flow driven by mode conversion in tokamak plasmas*, Physical Review Letters, 90 (2003).

[14] Peter Kogge, Keren Bergman, Shekhar Borkar, Dan Campbell, William Carlson, William Dally, Monty Denneau, Paul Franzon, William Harrod, Kerry Hill, Jon Hiller, Sherman Karp, Stephen Keckler, Dean Klein, Robert Lucas, Mark Richards, Al Scarpelli, Steven Scott, Allan Snavely, Thomas Sterling, R. Stanley Williams, and Katherine Yelick, *ExaScale computing study: Technology challenges in achieving exascale systems*, Tech. Report TR-2008-13, University of Notre Dame, May 2008. Report available at `http://www.cse.nd.edu/Reports/2008/TR-2008-13.pdf`.

[15] John E. Savage, *Extending the Hong-Kung model to memory hierarchies*, Computing and Combinatorics, 959/1995 (1995), pp. 270–281.

[16] Sivan Toledo, *Locality of reference in LU decomposition with partial pivoting*, SIAM Journal on Matrix Analysis and Applications, 18 (1997), pp. 1065–1081.

[17] Sivan Toledo, *A survey of out-of-core algorithms in numerical linear algebra*, in External memory algorithms, James M. Abello and Jeffrey Scott Vitter, eds., American Mathematical Society, Boston, MA, USA, 1999, pp. 161–179.

[18] Sivan Toledo and Fred G. Gustavson, *The design and implementation of SOLAR, a portable library for scalable out-of-core linear algebra computations*, in Proceedings of the fourth workshop on I/O in parallel and distributed systems: part of the federated computing research conference, IOPADS '96, New York, NY, USA, 1996, ACM, pp. 28–40.

[19] Robert A. van der Geijn, *Using PLAPACK: Parallel Linear Algebra Library*, MIT Press, USA, 1997.

[20] J. S. Vetter, R. Glassbrook, J. Dongarra, K. Schwan, B. Loftis, S. McNally, J. Meredith, J. Rogers, P. Roth, K. Spafford, and S. Yalamanchili, *Keeneland: Bringing heterogeneous gpu computing to the computational science community*, IEEE Computing in Science and Engineering, 5 (2011). `http://dx.doi.org/10.1109/MCSE.2011.83`.

[21] Jeffrey Scott Vitter, *External memory algorithms and data structures: Dealing with massive data*, ACM Comput. Surv., 33 (2001), pp. 209–271.

[22] Kwai L. Wong, Eduardo D'Azevedo, Harvy Hu, and Shiquan Su, *A performance study of solving a large dense matrix for radiation heat transfer*, Presented at the SIAM Conference on Computational Science and Engineering, Boston, Massachusetts, 2013.